SR-POD: Sample Rotation based on Principal-axis Orientation Distribution for Data Augmentation in Deep Object Detection

Yue Xi[1], Jiangbin Zheng[1*], Xiuxiu Li[2], Xinying Xu[3], Jinchang Ren[3,4*], Gang Xie[5]

[1] School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2] School of Computer Science, Xi'an University of Technology, Xi'an, China
[3]College of Electrical and Power Engineering, Taiyuan University of Technology, Taiyuan, China
[4] Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, UK
[5] College of Electronic Information Engineering, Taiyuan University of Science and Technology, Taiyuan, China

Abstract: Convolutional neural networks (CNNs) have outperformed most state-of-the-art methods in object detection. However, CNNs suffer the difficulty of detecting objects with rotation, because the dataset used to train the CCNs often does not contain sufficient samples with various angles of orientation. In this paper, we propose a novel data-augmentation approach to handle samples with rotation, which utilizes the distribution of the object's orientation without the time-consuming process of rotating the sample images. Firstly, we present an orientation descriptor, named as "principal-axis orientation'', to describe the orientation of the object's principal axis in an image and estimate the distribution of objects' principal-axis orientations (PODs) of the whole dataset. Secondly, we define a similarity metric to calculate the POD similarity between the training set and an additional dataset, which is built by randomly selecting images from the *benchmark ImageNet ILSVRC2012* dataset. Finally, we optimize a cost function to obtain an optimal rotation angle, which indicates the highest POD similarity between the two aforementioned data sets. In order to evaluate our data augmentation method for object detection, experiments, conducted on the *benchmark PASCAL VOC2007* dataset, show that with the training set augmented using our method, the average precision (AP) of the Faster RCNN in the TV-monitor is improved by *7.5%*. In addition, our experimental results also demonstrate that new samples generated by random rotation are more likely to result in poor performance of object detection.

Keywords*:* Data augmentation; Deep object detection; Deep learning; Object rotation.

***Corresponding authors:**

Prof. Jiangbin Zheng
School of Computer Science
Northwestern Polytechnical University
127 West Youyi Road
Xi'an, 710072
China

Email: zhengjb@nwpu.edu.cn
Tel: +86 13259476882

Dr Jinchang Ren
Department of Electronic and Electrical Engineering
University of Strathclyde
204 George Street
Glasgow, G1 1XW
United Kingdom

Email: jinchang.ren@strath.ac.uk
Tel: +44 141 5482384

1 INTRODUCTION

Convolutional neural networks (CNNs), typically trained on large scale of data, have significantly advanced the performance of the solutions to various vision problems such as object detection [3][6][7][15][30], salient object detection [34][36][37][40][41] and object recognition [5][18][26][31]. Performance of CNNs on object detection is largely attributed to networks with millions of parameters, which have a strong ability to extract rich, high-level object representation features. In order to ensure good performance, CNNs must be trained on large scale of data. Otherwise, if being trained on limited data they tend to lead into overfitting. Unfortunately, labelling a large number of images is not only time consuming and tedious, but also requires professional knowledge and skills[32]. Moreover, there are few (or no) training instances in some situations, such as a few-shot or zero-shot learning scenario**Error! Reference source not found.**[29]. To mitigate the issue to some extent, an effective technique called data augmentation, which generates real samples, is widely adopted to extend the training data**Error! Reference source not found.**[14].

Research has been devoted to data augmentation. Existing methods can be grouped into two categories, i.e., the guided methods and non-guided methods. In the guided methods such as in[11][19][21], external information (e.g., background, texture, noise) is introduced to build crafted feature descriptors or 3D models. Whereas the non-guided methods, such as **Error! Reference source not found.**[12][28] perform affine or non-linear transformation (e.g. flipping, cropping, rotating, adding noise) on existing samples. Compared with the guided methods, non-guided methods do not need to build complex data models, so they are easy to implement and also computationally less expensive. However, the non-guided approaches require certain prior knowledge of data; otherwise, they tend to produce fake samples, which do not exist in the real world. These samples, instead of enhancing the training, would slow down the training and lead to samples misclassified during training and testing.

Due to varying viewpoints and object orientations in 3D world, changes in object orientation of 2D image are ubiquitous [9]. In order to deal with object rotation, we propose a novel data augmentation method guided by the distribution of object orientation. Our work is based on a simple observation, i.e., when the distribution of object orientation of the training set is similar to that of the testing set, the training set can sufficiently cover the variability (i.e. object orientation) in the testing set. In addition, we integrate guided and non-guided methods into our framework, as shown in Fig.1. Specifically, firstly, we present an orientation descriptor to estimate the principal-axis orientation

distribution (POD) from dataset. Secondly, we design a similarity function to quantify the similarity of the POD between the training set and the testing set. Finally, a cost function is designed to enforce the POD of the training samples to be similar to that of the testing set after rotating. Moreover, we optimize the cost function to obtain an optimal rotation angle, at which the highest similarity is achieved between the two sets in terms of POD.

Our major contributions presented in this paper are in four folds: (1) We explore a novel data augmentation framework, which extending training set achieves a better coverage of objects' varying orientations in testing data, to improve the performance of CNNs on object detection; (2) We design a principal-axis orientation descriptor based on superpixel segmentation to represent the orientation of an object in an image; (3) We propose a similarity measure method of two datasets based on principal-axis orientation distribution; and (4) We evaluate the performance of CNNs on object detection with and without rotating images in testing set.

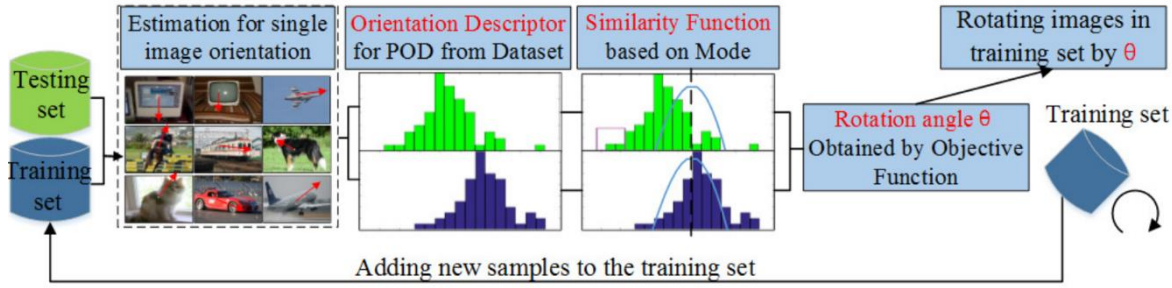


Fig.1. Illustration of our proposed data augmentation algorithm.

2 RELATED WORKS

The data augmentation technique extends training set through generating new samples with better coverage of variability in testing set. According to whether there is external data introduced, existing approaches can be grouped into non-guided methods and guided methods.

Non-guided methods mean that no external data is introduced during the generation of synthetic samples. Krizhevsky was the first to employ affine transformations (flipping, rotating and PCA-based intensity transformation) to increase the size of training set for learning deep models in order to avoid overfitting caused by a small set of training samples [12]. This method could effectively increase the variability in orientation and images' colour space in the training set. Zeiler et al [28] adopted it due to the low computational cost. Vincent et al. added Gaussian noise

to original images to improve the robustness of the deep learning models to perturbations [23][27]. Szegedy et al. proposed a random sample cropping method using scale and aspect ratio augmentation for object occlusion [21]. Zhong et al. proposed to crop images based on supervised data augmentation. Pasupa K and Sunhen W proposed a novel method based on key point detection, which extracted a set of feature points, connected them with each other to form the contour of an object. It was shown that this method could improve the variability in the contour of objects from training set, and avoid interference on classifiers from background and texture[16]. To sum up, these methods generate synthetic samples using a variety of transformations on original samples to improve the variability of dataset. The variability has the potential to improve the performance in hyperspectral imagery and video processing [35][38][39][42].

There have been a few approaches in the second category of guided methods. Seyyedsalehi et al. proposed a nonlinear manifold separator neural network, which generated new samples by extracting a variety of face expression and combining them, to solve the problem of variability in face expression [19]. Similarly, Lv and Shao [14] proposed a data augmentation method for facial data, by adding prior knowledge (such as hairstyle, glasses, lighting, etc.) to the training set, in order to reduce the influence of the changes in face expression and illumination. In order to extend training set, Charalambous and Bharath [2] proposed a guided method for gait recognition. The authors added a variety of confounding factors (such as hairstyle, clothing, etc.) to simulate gait video data. Furthermore, new samples can be produced from 3D models. Kittler et al. [11][20] generated 2D facial samples from a 3D facial model to extend the training set. It suggested that these methods were beneficial in some situations (e.g., a few-shot or zero-shot learning scenario) and a 3D model could be easily obtained. In Dixit et al. [5], a guided data augmentation approach is proposed in the feature space, which first learned the characteristics of existing training samples, and then produced samples with a series of desired attributes (such as depth or pose) by combining features. In summary, these methods extend the original dataset under the guidance of external data to increase coverage of variability.

In general, data augmentation has evolved from non-guided methods to guided methods. However, the key challenge is how to generate synthetic samples which can sufficiently and reasonably cover the variability in the real world. Existing research tackling variability mainly focuses on light, background, texture, etc. There is little research reported which looks at object orientation. In this work, we focus on object orientation and augment data to extend its coverage of various orientation angels. Marginally related to our proposed method, Peng et al. In [17] rendered 3D models with various factors (such as light, texture, background, etc.) and generated new samples to augment the

original dataset. Note that the key difference between this approach and our proposed method lies in the way that we extend the training set using rotation transformation to improve the coverage of object orientation in testing set. We illustrate this approach in more details in the next section.

3 OVERVIEW OF THE PROPOSED SYSTEM

As shown in Fig.1, we present a novel data augmentation method which is guided by the distribution of objects' orientation. We start with an orientation descriptor for object orientation estimation. An image is divided into a set of homogeneous regions and its orientation is jointly determined by the orientations of these homogeneous regions (Sec. 4.1). We then design a similarity measure to compare the distribution of the principle-axis orientations of the training and testing sets, which is robust to outliers (Sec. 4.2). Finally, we optimize a cost function with permutation matrix to compute the optimal rotation angle, at which the highest similarity between the two sets is achieved (Sec. 4.3).

4 THE PROPOSED SYSTEM

4.1 Estimating object orientation

The key to estimating object orientation is to design an orientation descriptor. The descriptor is based on an observation, that the object orientation is jointly determined by the orientations of its multiple homogeneous regions. Specifically, we use the principal axis, where the projection area is minimized, to describe the orientation of a region. Homogeneous regions can be obtained using superpixel segmentation techniques. We present a method based on superpixel segmentation to estimate the principal-axis orientation of an object. It is mainly composed of four stages as shown in Fig. 2.

Stage 1: the input image is over-segmented by the SLIC algorithm [1] which clusters pixels in colour space and image plane space to efficiently generate K compact and nearly uniform superpixels $R = \{R_1, \dots, R_k\}$. Formally, the distance function D_s is defined as follows:

$$D_s = d_{lab} + \frac{m}{s} d_{xy}$$

$$D_s = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} + \frac{m}{s}\sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \quad (1)$$

where $C_k=[l_k, a_k, b_k, x_k, y_k]^T$ denotes the centres of the superpixel clusters; d_{lab} is the Euclidean distance in CIELAB colour space and d_{xy} is the distance on image plane; D_s is the sum of d_{lab} and d_{xy}; $m \in [1,20]$ is introduced to control the compactness of a superpixel and s is the spatial interval of every superpixel, which offers a tradeoff between colour similarity and spatial proximity.

Stage 2: we analyse oriented patterns [10] to compute the gradient map of every superpixel R_i. In our work, we compute the gradient map of an image using the integral of the first derivative instead of the first derivative itself (like Sobel), because the integral of the first derivative provides a better stability when images suffer from low signal-noise ratio. Mathematically, the orientation $\theta(x,y)$ and magnitude $g(x,y)$ of gradient are defined as follows:

$$\theta(x,y)=\frac{1}{2}\tan^{-1}\frac{\iint_{\Omega}2I_xI_y dxdy}{\iint_{\Omega}(I_x^2-I_y^2)dxdy}+\frac{\pi}{2} \tag{2}$$

$$g(x,y)=\frac{\left(\iint_{\Omega}(I_x^2-I_y^2)dxdy\right)^2+\left(\iint_{\Omega}2I_xI_y dxdy\right)^2}{\left(\iint_{\Omega}(I_x^2+I_y^2)dxdy\right)^2} \tag{3}$$

where I_x and I_y are the partial derivative of the image I with respect to x and y respectively; tan^{-1} is a four quadrant inverse tangent function; Ω, $\theta(x,y)$ and $g(x,y)$ denote the neighbours of the pixel, the orientation and magnitude of the gradient of the pixel located at (x,y), respectively.

Stage 3: the principal-axis orientation β_i of the superpixel R_i is calculated from the histogram of gradient orientation $\theta(x,y)$ and magnitude $g(x,y)$ [13] as:

$$H_i(\alpha)=\sum_{\theta(x,y)\in\{\alpha-\frac{bin}{2},\alpha+\frac{bin}{2}\}} g(x,y) \tag{4}$$

where $\alpha=\{0,bin,bin\times 2,\ldots,\pi-bin\}$, $(x,y)\in R_i$, and $bin\in[0^\circ,180^\circ]$ is the sampling interval of pixel gradient orientation, as shown in Fig. 3. The gradient orientations of most pixels inside a superpixel are similar owing to the consistency of superpixel, resulting in a peak in a histogram. As a result, the angle β_i corresponding to the peak of the histogram $H_i(\alpha)$ is considered as the principal-axis of the super-pixel, which is defined as:

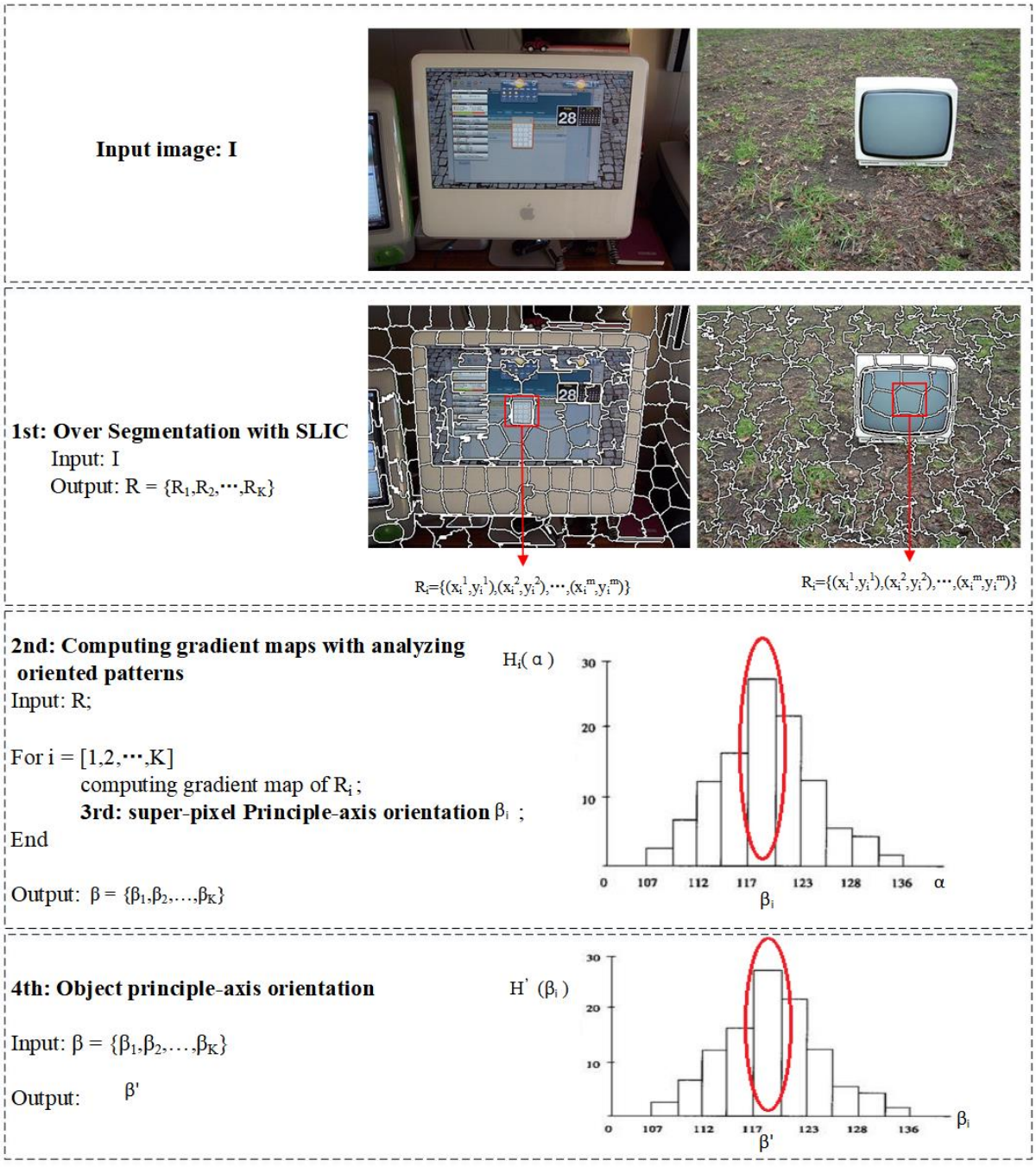


Fig.2. Estimating the orientation of an object based on superpixel segmentation

$$\beta_i = peak\left(H_i(\alpha)\right) \tag{5}$$

Stage 4, the orientation of the object's principal-axis, denoted as β', can be computed from the β_i and magnitude $H_i(\beta_i)$ of the superpixel R_i as:

$$H^{'}(\alpha) = \sum_{\beta_i \in \{\alpha - \frac{bin}{2}, \alpha + \frac{bin}{2}\}} H_i(\beta_i), \beta^{'} = peak(H^{'}(\alpha)) \tag{6}$$

where the angle $\beta^{'}$ is the orientation of the object.

4.2 Similarity measure based on mode

With the resultant principal-axis orientation, we may now define the similarity measure to quantify their POD similarity between the training and testing sets. However, since the reliability and stability of such similarity measure are vulnerable to outliers, we propose a novel similarity measure based on mode, which is a value that appears most often. The similarity measure process is mainly composed of two stages, i.e., mode estimation and similarity measure.

Stage 1: Estimating the coordinate of the mode. The histogram of the principal-axis orientation can provide an interval which a mode lies in. But it is difficult to accurately compute its mode owing to sampling interval. Therefore, we estimate the coordinate of the mode using an interpolation method as shown in Fig. 3.

$$M_0 = L + \frac{\Delta_1}{\Delta_1 + \Delta_2} \times d \quad or \quad M_0 = U - \frac{\Delta_2}{\Delta_1 + \Delta_2} \times d \tag{7}$$

where M_0 is the coordinate of a mode on x-axis of the histogram; L and U are the lower and upper bound of the interval where the mode lies in, respectively; Δ_1 is the difference between the value of the mode and L; Δ_2 is the difference between the value of the mode and U; d denotes the interval width.

Stage 2: Similarity measure. The task is to evaluate the difference between the orientation distributions of the training set and the testing set. We assume that the distribution difference belongs to the normal distribution where its mean value is the mode obtained in Stage 1 and its variance value is *1*.

$$S = \frac{1}{\sqrt{2\pi}} \times C \times \sum_{i=1}^{N} |y_i - y_i^{'}| \times e^{-\frac{1}{2}(x_i - \mu)^2} \tag{8}$$

where $S \in [0,1]$ is a function that makes a comparison between the principal-axis orientation distributions of the training and testing sets. $S = 1$ if two datasets are identical in terms of orientation distribution and $S = 0$ if they have nothing in common. y_i and x_i are the i-th value and interval of principal-axis orientation distribution from testing set, respectively; y' is the i-th value of orientation distribution of principal-axis in training set; μ is M_0 and $|\,.\,|$ is the *L1* norm; C is the normalized constant and N is the number of orientation bins.

The above formula is re-written in vector form to improve the computational efficiency. It is assumed that A is a diagonal matrix.

$$A=\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \tag{9}$$

$$\begin{bmatrix} e^{a_{11}} & 0 & \cdots & 0 \\ 0 & e^{a_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{a_{nn}} \end{bmatrix} = \begin{bmatrix} \sum_{k=0}^{+\infty} \frac{a_{11}^k}{k!} & 0 & \cdots & 0 \\ 0 & \sum_{k=0}^{+\infty} \frac{a_{22}^k}{k!} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sum_{k=0}^{+\infty} \frac{a_{nn}^k}{k!} \end{bmatrix} = \tag{10}$$

$$= \sum_{k=0}^{+\infty} \frac{1}{k!} \begin{bmatrix} a_{11}^k & 0 & \cdots & 0 \\ 0 & a_{22}^k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn}^k \end{bmatrix} = \sum_{k=0}^{+\infty} \frac{1}{k!} A^k = e^A \tag{11}$$

Therefore, $A = -\frac{1}{2} diag[(X-\mathbf{u})\times(X-\mathbf{u})^T]$

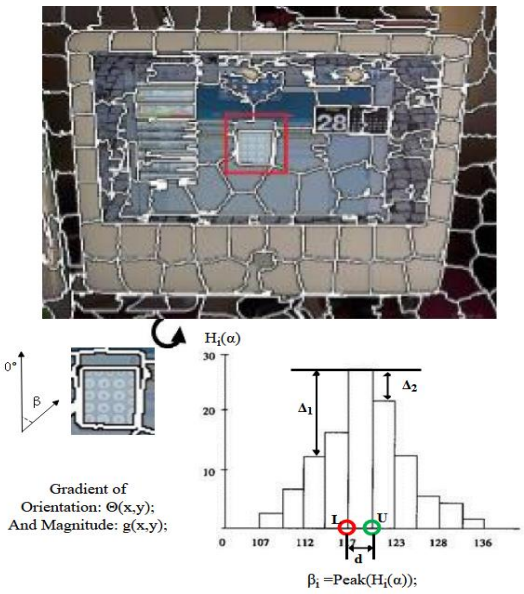


Fig.3. Estimation of superpixel orientation β_i and Coordinate estimation of mode in Hi(α) based on linear interpolation.

$$S = C \times |e^{A} \times (Y - Y^{'})| \tag{12}$$

where $X = [x_1, \ldots, x_N]^T, Y = [y_1, \ldots, y_N]^T, Y^{'} = [y_1^{'}, \ldots, y_N^{'}]^T, \mathbf{u} = [\mu, \ldots, \mu]^T$.

4.3 Rotating images with permutation matrix

Each image in the training set is clockwise rotated by Θ to form a new statistic on the distribution of the principal-axis' orientation. Permutation matrix is used to achieve cyclic shifts of the distribution [8]. An $n \times n$ permutation matrix **P**(**U**) is obtained from the $n \times 1$ vector **U** by concatenating all possible cyclic shifts of **U**:

$$Permutation matrix: \mathbf{P}(\mathbf{U}) = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{13}$$

where the first row is a vector **U**, the second row is **U** shifted the first element to the right (the last wraps around), and so on. Therefore, we propose a cost function to learn the parameter Θ:

$$\arg\min_{\Theta}(-S) = k \times |e^{A} \times (P^{\Theta} \times Y - Y^{'})| \tag{14}$$

In practice, the optimization problem can be solved by stochastic gradient descent (SGD), which has been widely applied to resolve complicated optimization problems. In conclusion, our algorithm is summarized in Algorithm 1.

Algorithm 1: the summarized final Algorithm

Input: training set I_{train} and testing set I_{test}.

- Estimate the orientation of single image in I_{train} and I_{test}.
- Compute the Principal-axis Orientation Distribution (POD) of I_{train} and I_{test}.
- Compute the POD similarity between I_{train} and I_{test}.
- Achieve rotation angle θ by Gradient Descent.

Output: rotate images in training set by θ.

5 EXPERIMENTAL RESULTS

We fist explore the effect of rotating the testing images on performance of CNNs in object detection, then we evaluate our similarity measure of principal-axis orientation distribution and eventually demonstrate our data augmentation for object detection.

We use the benchmark datasets *PASCAL VOC2007*, which contains 20 categories, 9963 RGB images and detailed annotations for about 24640 objects. Rotation matrix is performed on images and the operation is simple and computationally cheap. However, it may introduce much background clutter. In our method, we first resize bounding box which covers an object to a square; the maximum inscribed circle of the resized square is clockwise rotated by 0.5°; and finally, the square is resized to its original size. In this way, the number of the benchmark is increased from 9,963 to 7,173,360.

5.1 The impact of rotating testing images on the performance of CNNs in object detection

In this section, we explore the performance of Faster RCNN and RCNN on object detection, when testing images are rotated [6][18]. The experiment consists of four steps: i) we pre-train Faster RCNN and RCNN on the *ImageNet ILSVRC2012* challenge; ii) those pre-trained models are fine-tuned on *VOC2007* and the clockwise rotation angle θ of each image in testing set is initialized with 0°; iii) *VOC2007*-rotation consisting of images selected by rotation angle θ is tested using the fine-tuned Faster RCNN and RCNN; the detection result is measured according to the average precision (AP) and mean AP (mAP) over all object categories (20); and iv) the rotation angle θ is increased by *0.5°* and we repeat Steps iii and iv.

Faster RCNN experimental analysis. *First*, all of the 21 curves show approximate symmetry about *y = 180°*, as shown in Fig. 4; The mAP curve gradually decreases from 58.3% at 0° to 22.6% at 180° with a slight increase at around 90° and 180° (2.8% from 15.3% at 76° to 18.1% at 90°, 9.5% from 13.1% at 145° to 22.6% at 180°, respectively). *Second*, there is a sharp increase in the curve of TV-monitor at around 90° and 180° (18.6% from 13.8% at 62° to 32.4% at 92°, 23.9% from 10.6% at 148° to 34.5% at 180°), similar to some of other categories (i.e. cat, cow, dog, bird, sheep, aeroplane, motorbike and bicycle) as shown in Fig. 6, respectively. Because there is a lowest similarity in the distribution of the principal-axis's orientation between the training and testing set, if rotation angle is

62° or 148°. While there is a highest similarity in principal-axis orientation distribution between training and testing set, if images are rotated by 92° or 180°.

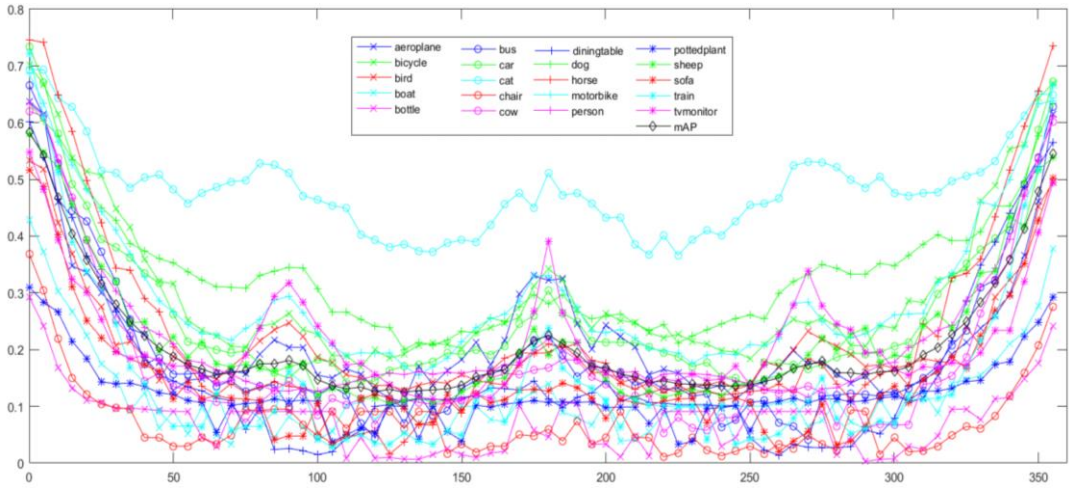


Fig.4. The object detection results of the Faster RCNN on VOC2007. The X axis indicates the rotation angle ranging from 0° to 359° with an interval of 0.5°, and the Y axis represents the resultant AP. The mAP is obtained over all object categories with the Faster RCNN.

RCNN experimental analysis. *First*, similar to the Faster RCNN, all of the 21 curves show approximate symmetry about y = 180° ; mAP curve is gradually decreased from 54.1% at 0° to 21.47% at 180° with a slight increase at around 90° and 180° (6.04% from 8.88% at 57° to 14.92% at 76° , 12.02% from 9.45% at 162° to 21.47% at 180° , respectively). *Second*, there is a sharp increase in the curve of TV-monitor at around 90° and 180° (15.56% from 6.45% at 46° to 22.01% at 99°, 30.83% from 6.15% at 146° to 36.98% at 180°), similar to some of other categories (i.e., cat, cow, dog, bird, sheep, aeroplane, motorbike and bicycle) as shown in Fig. 5.

Experimental results show that rotating images in the testing set can affect the detection performance of deep learning models. However, when the rotation angle is around 90° or 180°, there is a significant boost in AP. The features extracted by CNN are sensitive to image rotation. If the test image is rotated, the features extracted from the rotated image would be different from the ones extracted from the original image. As a result, the classifier trained using samples without rotating has produced poor performance on our test set. In addition, many images in our test set have similar texture appearance after they are rotated by 90° or 180°. See Fig. 2 for example, the rectangle television still appears as a rectangle when it is rotated by 90° or 180°. Therefore, the features extracted are similar to the ones extracted from images in the original training set, which can be classified with higher accuracy.

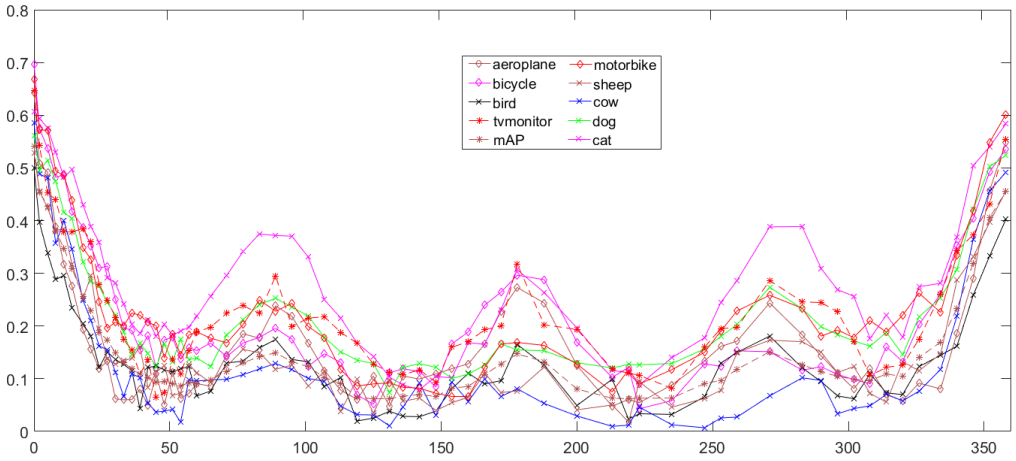


Fig.5. The object detection result obtained with the RCNN. The X axis indicates the rotation angle ranging from 0° to 359° with an interval of the testing set is 0.5°, and the Y axis represents resulted AP. Again mAP is obtained from all object categories.

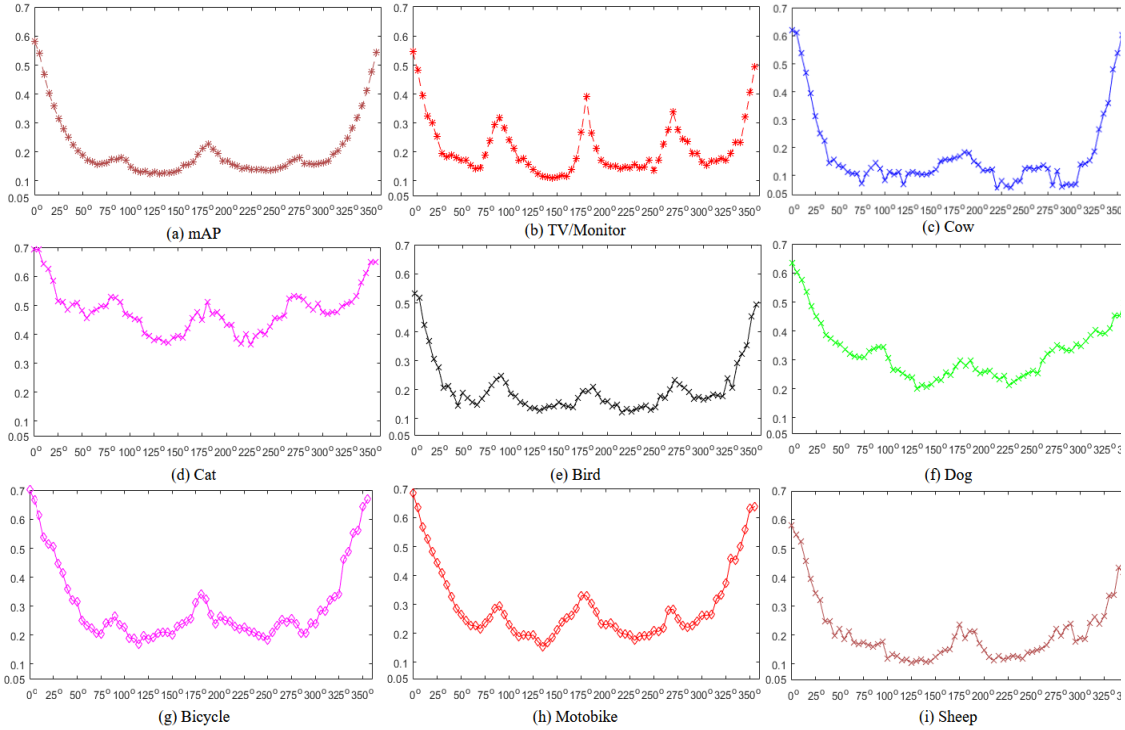
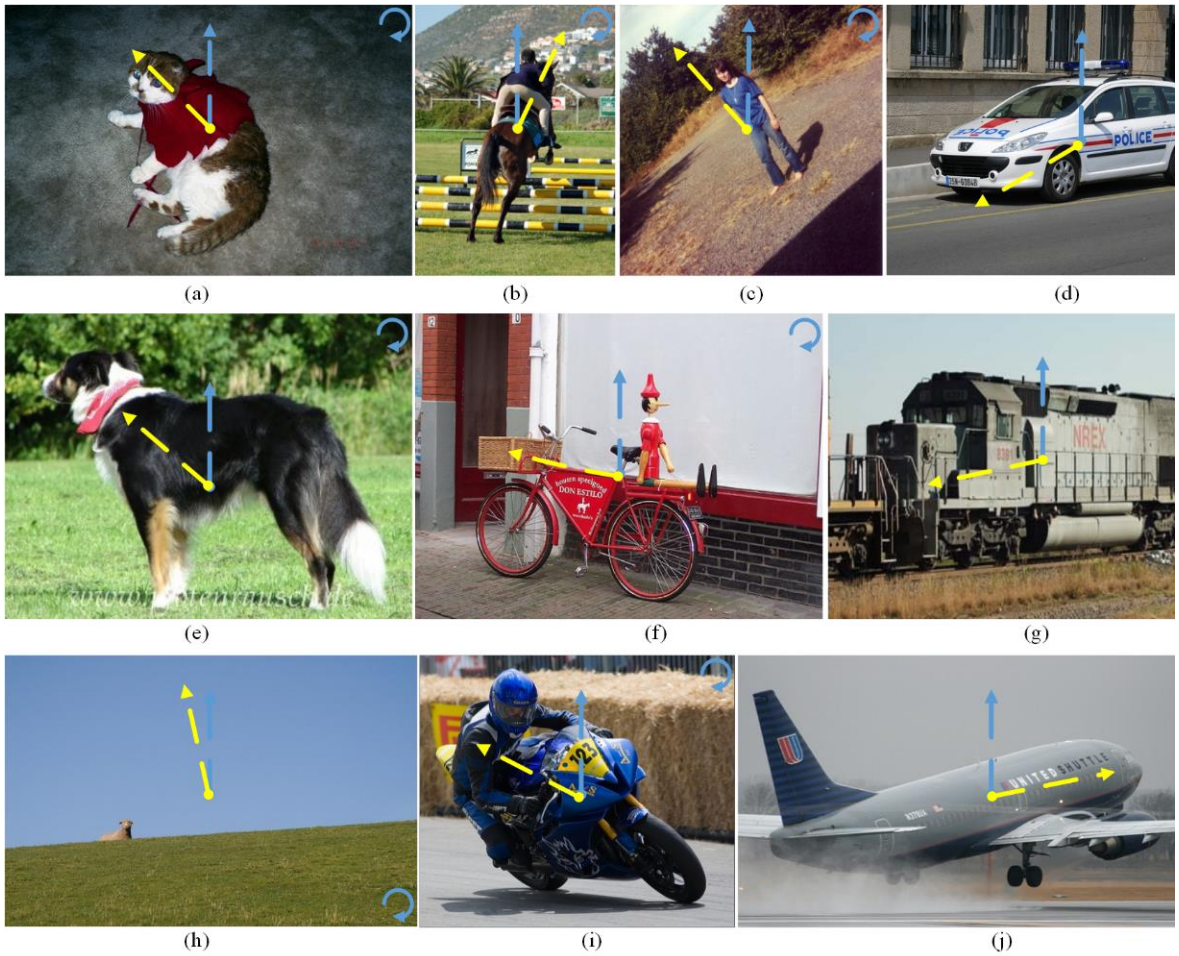


Fig.6. The detection mAP and AP obtained with the Faster RCNN over the categories of TV-Monitor, Cow, Cat, Bird, Dog, Bicycle, Motorbike and Sheep. The X axis indicates that the rotation angle ranging from 0° to 359° with an interval of 0.5°, and the Y axis represents the AP. The mAP is again obtained over all categories with the Faster RCNN.

5.2 Estimation of object orientation and comparison with existing method

To the best of our knowledge, there is no dataset for estimating object orientation[22]. So we test our proposed algorithm on a dataset which consists of 4896 images from the benchmark *PASCALVOC 2007*. 15 experts in image

processing are invited to label each object image. The estimation error is the difference between the estimated orientation and the ground truth. It is considered a correct prediction if the estimation error is less than 10°. The accuracy is 85.45%. Some of results are shown as in Fig. 7. Our method can estimate image orientation at clutter background (Fig. 7 (b) and (f)). Existing methods [4][22][24][25] just classified images into several distinguished orientations. So, these fail to estimate accurately image orientation.



(a) (b) (c) (d)

(e) (f) (g)

(h) (i) (j)

Fig.7. Results of our method for estimating object orientation. (a) -45.13° (b) 35.75° (c) 43.13° (d) -120.35° (e) -45.68° (f)-81.39° (g) -100.27° (h) -15.17° (i) -78.08° (j) 81.15°. The yellow arrows in each image represent the orientation which our method estimated.

5.3 Similarity measure of principal-axis orientation distribution

We analyze the distribution of objects' orientation in the training and testing sets using the principal-axis orientation based on superpixel. On the other hand, the formal experiment demonstrated that there is a strong relationship between TV-monitor and image rotation. So, the goal of this experiment is to verify that our method can significantly boost

the TV-monitor similarity between distribution of principal-axis orientation in the two sets. The experiment consists of these steps: 1) we perform similarity measure on *VOC2007* and achieve similarity between training and testing sets, 2) images generated using data augmentation are added to training set, 3) we also perform similarity measure on augmented dataset and achieve similarity between new training and testing sets.

Before data augmentation, the principal-axis orientation distribution sampling from training and testing set is shown as in Fig. 8 (a) and (b). After data augmentation, the distribution is shown in Fig. 8 (b) and (c). Experimental results show that the similarity is increased from 56.72% to 64.85%.

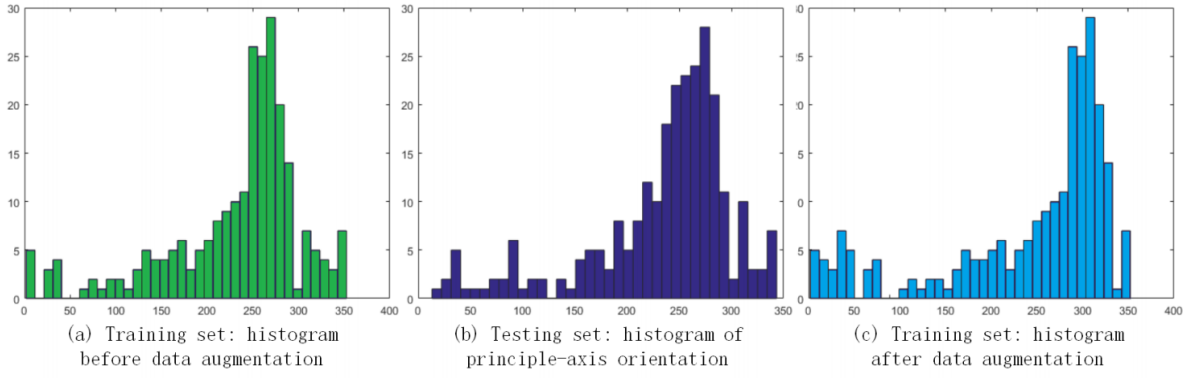


Fig.8. The distribution of the principal-axis orientation.

5.4 Data augmentation applied to object detection

We verify that data augmentation can boost the detection performance of Faster RCNN. Images in the training set are flipped horizontally, which can reduce parameter space of rotation angle of the training set from 360° to 180°. For this experiment, we follow four steps: 1) Faster RCNN is pre-trained on *ImageNet ILSVRC2012*, fine-tuned on *VOC2007* and then tested on the testing set to obtain the AP and mAP, 2) the clockwise-rotation angle θ of each training image is initialized with 0°, 3) new training set contains images selected by θ; we also fine-tune the pre-trained Faster RCNN on the new training set and then run it on the testing set in VOC2007 to achieve AP and mAP, 4) Increase θ by 3° and repeat Steps 3 and 4.

Experimental results show that when training images are rotated by 23°, the improvement in AP of the TV-monitor is highest, as shown in Fig. 9. The degree 23° approximately equals the value based on our theory. Moreover, there is a decline in mAP, when randomly rotated images are added to the training set. Because those samples do not actually exist in the real world and have misled classifiers.

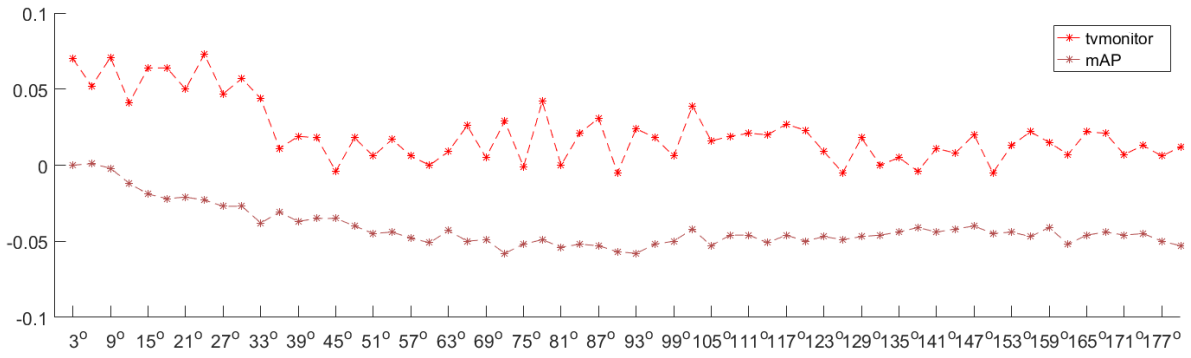


Fig.9. Data augmentation in training set. The X axis indicates that the rotation angle interval is 3° in the range of [0° , 180°]. The Y axis represents the AP and mAP of the Faster RCNN achieved over the categories of "TV-monitor".

6 CONCLUSION

In this paper, we proposed a simple and effective approach of data augmentation for CNNs based object detection. By analyzing the distribution of our proposed principal-axis orientation of the augmented training set, the optimal rotation angle can be learned at which the highest accuracy can be achieved. Our experiments on benchmark data set PASCAL VOC 2007 have demonstrated the effectiveness of our idea. Using the training set augmented by our sample rotation method, the average precision (AP) of Faster RCNN in TV-monitor has been improved by 7.5%. In addition, the proposed method can also be used for other detectors that require larger, more diverse training data to further improve their performance.

Two aspects are extremely interesting for future work. First, we have demonstrated that our data augmentation of principal-axis orientation can effectively cover the variability of object orientation in testing set. It would be interesting to design descriptors of other cues (i.e. texture, noise and occlusion) to further cover the variability in testing set. Secondly, this paper has demonstrated how to introduce external data to effectively augment the original dataset. It would be interesting to seek some worthy samples from the new data set to achieve a few-shot learning.

7 COMPLIANCE WITH ETHICAL STANDARDS

7.1 Funding

This work has been supported by Remote sensing information processing project (No. XXXXXX) and also the National Natural Science Foundation of China under the grant No.61502382.

7.2 Conflict of Interest

The authors declare that they have no conflict of interest.

7.3 Ethical Approval

가 This article does not contain any studies with human participants or animals performed by any of the authors.

REFERENCES

[1] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. IEEE transactions on pattern analysis and machine intelligence, 34(11), 2274-2282.

[2] Charalambous, C. C., & Bharath, A. A. (2016). A data augmentation methodology for training machine/deep learning gait recognition algorithms. arXiv preprint arXiv:1610.07570.

[3] Cheng, G., & Han, J. (2016). A survey on object detection in optical remote sensing images. ISPRS Journal of Photogrammetry and Remote Sensing, 117, 11-28.

[4] Ciocca, G., Cusano, C., & Schettini, R. (2015). Image orientation detection using LBP-based features and logistic regression. Multimedia Tools and Applications, 74(9), 3013-3034.

[5] Dixit, M., Kwitt, R., Niethammer, M., & Vasconcelos, N. (2017). AGA: Attribute-Guided Augmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7455-7463).

[6] Girshick, R. (2015). Fast r-cnn. In Computer Vision, 2015 IEEE International Conference on (pp. 1440-1448). IEEE.

[7] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 580-587).

[8] Henriques, J. F., Caseiro, R., Martins, P., & Batista, J. (2012). Exploiting the circulant structure of tracking-by-detection with kernels. In European Conference on Computer Vision (pp. 702-715). Springer, Berlin, Heidelberg.

[9] Jaderberg, M., Simonyan, K., & Zisserman, A. (2015). Spatial transformer networks. In Advances in Neural Information Processing Systems (pp. 2017-2025).

[10] Kass, M., & Witkin, A. (1986). Analyzing oriented patterns. Computer Vision, Graphics, and Image Processing, 36(1), 133.

[11] Kittler, J., Huber, P., Feng, Z. H., Hu, G., & Christmas, W. (2016). 3D morphable face models and their applications. In International Conference on Articulated Motion and Deformable Objects (pp. 185-206). Springer, Cham.

[12] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems (pp. 1097-1105).

[13] Liu, K., Skibbe, H., Schmidt, T., Blein, T., Palme, K., Brox, T., & Ronneberger, O. (2014). Rotation-invariant HOG descriptors using Fourier analysis in polar and spherical coordinates. International Journal of Computer Vision, 106(3), 342-364.

[14] Lv, J. J., Shao, X. H., Huang, J. S., Zhou, X. D., & Zhou, X. (2017). Data augmentation for face recognition. Neurocomputing, 230, 184-196.

[15] Ning, Q., Zhu, J., & Chen, C. (2018). Very Fast Semantic Image Segmentation Using Hierarchical Dilation and Feature Refining. Cognitive Computation, 10(1), 62-72.

[16] Pasupa, K., & Sunhem, W. (2016). A comparison between shallow and deep architecture classifiers on small dataset. In Information Technology and Electrical Engineering, 2016 8th International Conference on (pp. 1-6). IEEE.

[17] Peng, X., Sun, B., Ali, K., & Saenko, K. (2015). Learning deep object detectors from 3d models. In Computer Vision, 2015 IEEE International Conference on (pp. 1278-1286). IEEE.

[18] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems (pp. 91-99).

[19] Seyyedsalehi, S. Z., & Seyyedsalehi, S. A. (2014). Simultaneous learning of nonlinear manifolds based on the bottleneck neural network. Neural processing letters, 40(2), 191-209.

[20] Song, X., Feng, Z. H., Hu, G., Kittler, J., Christmas, W., & Wu, X. J. (2016). Dictionary Integration using 3D Morphable Face Models for Pose-invariant Collaborative-representation-based Classification. arXiv preprint arXiv:1611.00284.

[21] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., & Anguelov, D. & Rabinovich, A.(2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-9).

[22] Vailaya, A., Zhang, H., Yang, C., Liu, F. I., & Jain, A. K. (2002). Automatic image orientation detection. IEEE Transactions on Image Processing, 11(7), 746-755.

[23] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P. A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 11(Dec), 3371-3408.

[24] Wang, Y., & Zhang, H. (2001). Content-based image orientation detection with support vector machines. In Content-Based Access of Image and Video Libraries, 2001. IEEE Workshop on (pp. 17-23). IEEE.

[25] Wang, Y. M., & Zhang, H. (2004). Detecting image orientation based on low-level visual content. Computer Vision and Image Understanding, 93(3), 328-346.

[26] Wen, G., Hou, Z., Li, H., Li, D., Jiang, L., & Xun, E. (2017). Ensemble of deep neural networks with probability-based fusion for facial expression recognition. Cognitive Computation, 9(5), 597-610.

[27] Xie, J., Xu, L., & Chen, E. (2012). Image denoising and inpainting with deep neural networks. In Advances in Neural Information Processing Systems (pp. 341-349).

[28] Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In European Conference on Computer Vision (pp. 818-833). Springer, Cham.

[29] Zhang, S., Huang, K., Zhang, R., & Hussain, A. (2018). Learning from few samples with memory network. Cognitive Computation, 10(1), 15-22.

[30] Zhang, X., Yang, Y. H., Han, Z., Wang, H., & Gao, C. (2013). Object class detection: A survey. ACM Computing Surveys , 46(1), 10.

[31] Zheng, J., Xi, Y., Feng, M., Li, X., & Li, N. (2016). Object detection based on BING in optical remote sensing images. In Image and Signal Processing, BioMedical Engineering and Informatics, International Congress on (pp. 504-509). IEEE.

[32] Zhou, Z., Shin, J., Zhang, L., Gurudu, S., Gotway, M., & Liang, J. (2017). Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7340-7349).

[33] Han, J., Zhang, D., Cheng, G., Guo, L., & Ren, J. (2015). Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning. IEEE Transactions on Geoscience and Remote Sensing, 53(6), 3325-3337.

[34] Han, J., Zhang, D., Hu, X., Guo, L., Ren, J., & Wu, F. (2015). Background prior-based salient object detection via deep reconstruction residual. IEEE Transactions on Circuits and Systems for Video Technology, 25(8), 1309-1321.

[35] Zhao, C., Li, X., Ren, J., & Marshall, S. (2013). Improved sparse representation using adaptive spatial support for effective target detection in hyperspectral imagery. International Journal of Remote Sensing, 34(24), 8669-8684.

[36] Wang, Z., Ren, J., Zhang, D., Sun, M., & Jiang, J. (2018). A deep-learning based feature hybrid framework for spatiotemporal saliency detection inside videos. Neurocomputing, 287, 68-83.

[37] Yan, Y., Ren, J., Sun, G., Zhao, H., Han, J., Li, X., ... & Zhan, J. (2018). Unsupervised image saliency detection with Gestalt-laws guided optimization and visual attention based refinement. Pattern Recognition, 79, 65-78.

[38] Sun, G., Zhang, A., Ren, J., Ma, J., Wang, P., Zhang, Y., & Jia, X. (2017). Gravitation-Based Edge Detection in Hyperspectral Images. Remote Sensing, 9(6), 592.

[39] Yan, Y., Ren, J., Zhao, H., Sun, G., Wang, Z., Zheng, J., ... & Soraghan, J. (2018). Cognitive fusion of thermal and visible imagery for effective detection and tracking of pedestrians in videos. Cognitive Computation, 10(1), 94-104.

[40] Huang, R., Feng, W., & Sun, J. (2017). Color feature reinforcement for cosaliency detection without single saliency residuals. IEEE Signal Processing Letters, 24(5), 569-573.

[41] Cao, X., Tao, Z., Zhang, B., Fu, H., & Feng, W. (2014). Self-adaptively weighted co-saliency detection via rank constraint. IEEE Transactions on Image Processing, 23(9), 4175-4186.

[42] Feng, W., & Liu, Z. Q. (2008). Region-level image authentication using Bayesian structural content abstraction. IEEE Transactions on Image Processing, 17(12), 2413-2424.